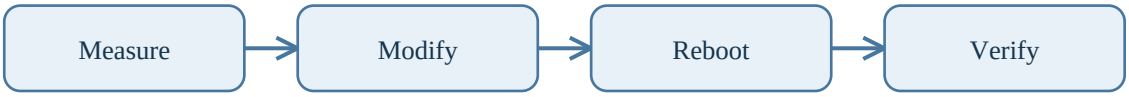
PP4MNK Balanced Optimization Handbook

EasyOS / Devuan Excalibur - Bilingual practical guide (English and Spanish)

What this handbook is
A step-by-step manual explaining the exact modifications used in this tuning project, why each one exists, how to apply it safely, what results to expect, and whether it is essential or optional. It is written for beginners first, but it also preserves the exact scripts for advanced users.

Safe tuning cycle

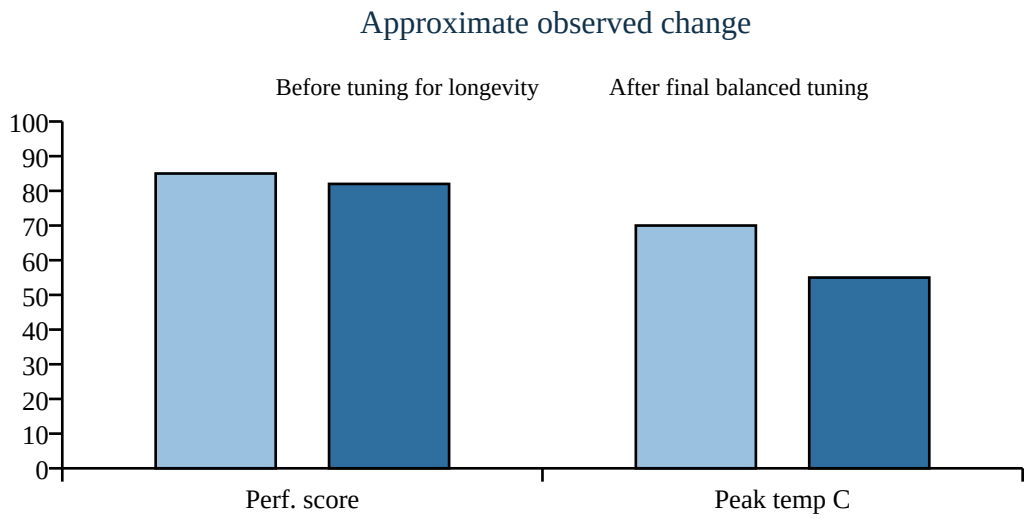


This document focuses on a practical goal: not maximum benchmark speed, but the best balance between responsiveness, lower temperature, fewer disk writes, and longer hardware life.

Part I - English guide

1. The idea behind the final configuration

After repeated tests during Easter week, the final decision was to prefer a balanced setup over an aggressive one. In practice, the CPU was capped at 2.8 GHz instead of allowing full turbo all the time, network tuning was made persistent through startup scripts, RAM and temporary storage were favored over disk writes, and the kernel networking stack was tuned for a 1 Gbps home connection. The result was not the absolute highest benchmark score, but a machine that stayed cooler, felt stable, and remained very responsive in everyday browsing, streaming and general desktop use.



Approximate observed trade-off during testing: a small loss in peak performance for a substantial reduction in heat.

Typical observed benefits
Lower CPU temperature under load; fewer unnecessary disk writes; smoother network behaviour; faster DNS resolution; consistent boot-time application of settings; and simpler maintenance because every change lives in a small, readable file in Startup.

2. Before you start

- Make a backup of any file you are going to replace. Beginners should never edit a system file without keeping a copy of the original.
- Apply one change at a time when possible. Reboot and test after each major step.
- If something goes wrong, disable the startup script by removing its executable bit or moving it out of Startup.
- These changes are aimed at a system similar to this one: EasyOS / Devuan-based, SSD storage, enough RAM, and a wired Ethernet connection.

File	Importance	Main goal
Zram-swap.sh	Recommended	Fast compressed swap in RAM
bootlocal.sh	Optional but useful	Move temporary writes from disk to RAM
pp4mnk-sysctl-loader	Essential if using sysctl.conf	Load kernel tuning at boot

File	Importance	Main goal
sysctl.config	Recommended	Tune TCP buffers and VM behaviour
pp4mnk-network-optimize	Optional	Enable NIC offloads and longer TX queue
pp4mnk-MTU-tunning	Recommended on PPPoE/VPN links	Apply 1492 / 1400 at boot
pp4mnk-dns-loader-v3	Optional	Set fast public DNS with quick timeouts
powertop-autotune	Optional	Apply energy-saving tunables after boot

3. Where to place the files

On this system, most persistent boot actions are simple shell scripts placed in **/root/Startup**. When the desktop starts, EasyOS runs the executable files in that directory. This makes the setup easy to understand: one file per tweak, one purpose per file.

Recommended workflow for a beginner

1. Open a terminal.
2. Go to the startup directory: `cd /root/Startup`
3. Create a script file with a text editor, for example Geany.
4. Paste the code exactly as shown in this document.
5. Save the file with the same name.
6. Make it executable: `chmod +x filename`
7. Reboot and verify the result.

> **Very important**
> A script only runs automatically if it is executable. If you save the file but forget `chmod +x`, the tweak will not be applied at boot.

4. CPU strategy: why reducing maximum frequency can be the smart choice

The most important practical decision in this tuning project was to limit the CPU to 2.8 GHz. This was not done because the machine was too slow. It was done because the hardware is older, and lower clock speed means less heat, less sustained voltage pressure, and usually much quieter operation. In normal desktop tasks, the difference between 2.8 GHz and full turbo is often small. In temperature, however, the difference can be large.

Setting	Balanced mode	Reason
scaling_max_freq	2800000	Keep the CPU cool and reduce stress
up_threshold	50	Let the CPU react quickly when load appears
powersave_bias	0	Keep responsiveness high within the chosen frequency cap

This handbook does not automate the CPU cap itself because that code was discussed separately, but the rest of the document assumes the same philosophy: trade a little peak speed for better thermals and a friendlier system.

5. Script-by-script implementation

Zram-swap.sh

Creates a compressed swap device in RAM. This is much faster than disk swap and reduces write activity on the SSD.

How to apply it - beginner steps
1. Create the file in /root/Startup.
2. Paste the script exactly.
3. Run chmod +x Zram-swap.sh.
4. Reboot.
5. Verify with free -h and swapon --show.

```
#!/bin/sh
echo "Activando ZRAM swap..."
modprobe zram
echo lzo-rle > /sys/block/zram0/comp_algorithm
echo 4G > /sys/block/zram0/disksize
mkswap /dev/zram0
swapon /dev/zram0
echo "ZRAM swap activado."
free -h
```

How to verify: Expected result: free -h shows swap active on zram; the system remains responsive under memory pressure.

bootlocal.sh

Moves temporary activity such as /tmp, logs, and browser cache toward RAM so the live session writes less to disk.

How to apply it - beginner steps
1. Create the file or edit your existing bootlocal.sh.
2. Paste the code.
3. Make it executable.
4. Reboot.
5. Verify with mount | grep /tmp and mount | grep /var/log.

```
#!/bin/sh
sysctl -w vm.swappiness=10
swapoff -a
if ! mount | grep -q "/tmp type tmpfs"; then
    mount -t tmpfs -o size=1G tmpfs /tmp
fi
mkdir -p /tmp/log
mount --bind /tmp/log /var/log
mkdir -p /tmp/browser_cache
```

How to verify: Expected result: /tmp is tmpfs, logs live under /tmp/log, and browser cache can be redirected to RAM.

pp4mnk-sysctl-loader

Loads the values from the kernel tuning file automatically at every boot.

How to apply it - beginner steps

1. Place the script in /root/Startup.
2. Make it executable.
3. Ensure the target sysctl file is in the correct location.
4. Reboot.
5. Verify with sysctl net.ipv4.tcp_fastopen or another tuned value.

```
#!/bin/sh
# Apply sysctl settings at every boot
sysctl -p >/dev/null 2>&1
```

How to verify: Expected result: the chosen kernel tuning values appear automatically after each reboot.

sysctl.config

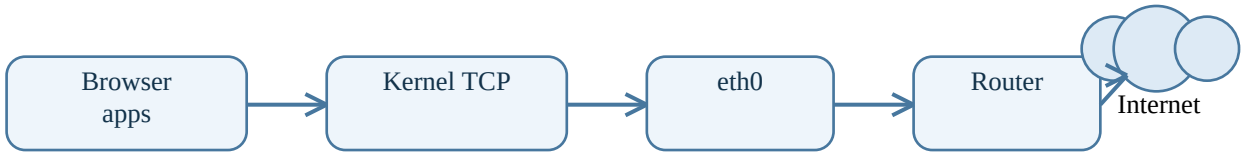
Contains the real kernel networking and memory values. This is the heart of TCP and VM tuning.

How to apply it - beginner steps

1. Create the file with the listed values.
2. Copy it to /etc/sysctl.conf or source it explicitly from your loader.
3. Save the file.
4. Reboot or run sysctl -p manually.
5. Verify each value with sysctl name_of_value.

```
# Max buffers
net.core.rmem_max=67108864
net.core.wmem_max=67108864

# TCP buffers
net.ipv4.tcp_rmem=4096 87380 67108864
net.ipv4.tcp_wmem=4096 65536 67108864

# Congestion control
net.ipv4.tcp_congestion_control=cubic

# Stability and efficiency
net.ipv4.tcp_fastopen=3
net.ipv4.tcp_mtu_probing=1
net.ipv4.tcp_slow_start_after_idle=0

# Network queue
net.core.netdev_max_backlog=5000

# RAM tuning
vm.swappiness=0
vm.vfs_cache_pressure=25
vm.dirty_ratio=10
vm.dirty_background_ratio=5
```

How to verify: Expected result: improved network buffering, quicker recovery from idle, less eagerness to swap, and more controlled dirty page behaviour.



sysctl buffers, MTU, offloading and DNS all act on this path

The kernel networking values operate mainly in the path shown above.

pp4mnk-network-optimize

Enables NIC offload features and raises txqueuelen so the interface can handle bursts more smoothly.

How to apply it - beginner steps
1. Install ethtool if missing.
2. Create the script in /root/Startup.
3. Make it executable.
4. Reboot.
5. Verify with ethtool -k eth0 and ifconfig eth0.

```
#!/bin/sh
ethtool -K eth0 tso on
ethtool -K eth0 gso on
ethtool -K eth0 sg on
ifconfig eth0 txqueuelen 5000
```

How to verify: Expected result: offload features show as enabled; txqueuelen becomes 5000.

Beginner warning
Offloading is not always a guaranteed improvement. If performance gets worse or video playback stutters, disable this script and test again.

pp4mnk-MTU-tunning

Applies MTU 1492 after a short delay so the network stack uses the right packet size.

How to apply it - beginner steps
1. Create the script.
2. Make it executable.
3. Reboot.
4. Verify with ip link show eth0.

```
#!/bin/sh
sleep 2
ip link set eth0 mtu 1492
```

How to verify: Expected result: eth0 reports MTU 1492.

pp4mnk-dns-loader-v3

Rewrites resolv.conf with the preferred DNS servers and resolver options.

How to apply it - beginner steps
1. Create the script.
2. Make it executable.
3. Reboot.
4. Check /etc/resolv.conf.
5. Test with nslookup or ping.

```
#!/bin/sh
RESOLVCONF="/etc/resolv.conf"
LOGFILE="/tmp/dns-loader.log"

cat > "$RESOLVCONF" <<EOF
nameserver 1.0.0.1
nameserver 1.1.1.1
nameserver 8.8.8.8
options timeout:1 attempts:2 single-request-reopen
EOF
```

How to verify: Expected result: resolv.conf contains Cloudflare and Google DNS plus short timeout options.

Beginner warning
Your network manager or DHCP client may rewrite `/etc/resolv.conf`. If your DNS changes disappear after a reboot or reconnection, that is the first place to investigate.

powertop-autotune

Waits until the system finishes booting and then asks powertop to apply its power-saving recommendations.

How to apply it - beginner steps
1. Install powertop if missing.
2. Create the script.
3. Make it executable.
4. Reboot.
5. Verify inside powertop or by checking tunables manually.

```
#!/bin/sh
sleep 20
powertop --auto-tune
```

How to verify: Expected result: a number of power-saving tunables move to Good after boot.

6. Verification commands

What to check	Command	What good looks like
CPU cap	cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq	2800000 if using the balanced cap
MTU	ip link show eth0	mtu 1492
DNS	cat /etc/resolv.conf	1.0.0.1 / 1.1.1.1 / 8.8.8.8 plus resolver options
sysctl	sysctl net.ipv4.tcp_fastopen	value = 3
zram swap	swapon --show	/dev/zram0 appears as active swap
tmp in RAM	mount | grep "/tmp type tmpfs"	/tmp mounted as tmpfs
offloads	ethtool -k eth0	TSO/GSO/SG enabled if script works

7. What matters most, and what is optional

Strongest practical impact	Medium impact	Optional / test first
CPU frequency strategy and thermal behaviour	sysctl tuning	NIC offloads
MTU set correctly for the line	ZRAM swap	Custom DNS loader
Startup persistence (scripts really run)	/tmp/logs/browser cache in RAM	powertop autotune

A beginner-friendly conclusion
If you only copy three ideas from this handbook, make them these: use the correct MTU for your line, keep the CPU cooler if you do not need maximum turbo, and make every change easy to undo. Those three habits give most of the benefit with much less risk.

Parte II - Guía en español

1. Idea general de la configuración final

Tras muchas pruebas, la decisión final fue priorizar el equilibrio. En vez de buscar la puntuación más alta posible, la configuración se orientó a mantener el equipo más fresco, reducir escrituras innecesarias en disco, hacer persistentes los ajustes de red y conservar una sensación de sistema ágil en el uso diario. Esa filosofía es especialmente útil en hardware veterano: un poco menos de velocidad punta puede traducirse en mucha menos temperatura y menos castigo continuo para la CPU y el SSD.

Qué se gana con este enfoque
Menos temperatura bajo carga, menos ruido, menos escrituras al SSD, una red más estable y una configuración más fácil de mantener porque cada cambio está guardado en un script pequeño y legible.

2. Antes de tocar nada

- Haz copia de seguridad de cualquier archivo que vayas a modificar.
- Aplica un cambio cada vez y reinicia para comprobarlo.
- Si algo falla, desactiva el script quitándole permisos de ejecución o sacándolo de /root/Startup.
- Estas optimizaciones tienen más sentido en un sistema parecido al descrito: EasyOS o Devuan, SSD, suficiente RAM y conexión Ethernet.

Archivo	Importancia	Objetivo principal
Zram-swap.sh	Recomendable	Swap comprimida y rápida en RAM
bootlocal.sh	Opcional pero útil	Mover actividad temporal del disco a la RAM
pp4mnk-sysctl-loader	Esencial si usas sysctl.conf	Cargar afinación del kernel al arrancar
sysctl.config	Recomendable	Ajustar buffers TCP y memoria virtual
pp4mnk-network-optimize	Opcional	Activar offloads y ampliar la cola TX
pp4mnk-MTU-tunning	Recomendable en líneas PPPoE	Aplicar MTU 1492 al inicio
pp4mnk-dns-loader-v3	Opcional	Fijar DNS rápidos con timeouts cortos
powertop-autotune	Opcional	Aplicar recomendaciones de ahorro energético

3. Dónde poner los archivos

En esta configuración, la mayoría de acciones persistentes se hacen con scripts ejecutables dentro de **/root/Startup**. EasyOS los lanza al iniciar el escritorio. Eso hace que todo sea muy fácil de seguir: un archivo por ajuste y cada archivo con una misión concreta.

Pasos sencillos para un usuario novato

1. Abre un terminal.
2. Entra en el directorio: `cd /root/Startup`
3. Crea el archivo con un editor como Geany.
4. Pega el código exactamente igual que aparece en este manual.
5. Guarda el archivo con el mismo nombre.
6. Dale permiso de ejecución: `chmod +x nombre_del_archivo`
7. Reinicia y comprueba el resultado.

> **Muy importante**
> Si el archivo no tiene permiso de ejecución, el ajuste no se aplicará al arrancar. Este detalle es una de las causas más comunes de error cuando alguien copia scripts por primera vez.

4. Estrategia de CPU: por qué limitar la frecuencia puede ser mejor

Una de las decisiones clave fue limitar la CPU a 2,8 GHz. No se hizo porque el equipo fuese lento, sino porque así trabaja más fresca y con menos estrés térmico. En tareas normales como navegar, ver streaming o usar el escritorio, la diferencia de sensación entre 2,8 GHz y el turbo máximo suele ser pequeña. En temperatura, sin embargo, la diferencia puede ser grande. Por eso esta estrategia tiene mucho sentido en un ordenador antiguo que se quiere conservar.

Zram-swap.sh

Crea una swap comprimida en RAM. Es mucho más rápida que la swap en disco y reduce escrituras en el SSD.

Cómo aplicarlo paso a paso
1. Crea el archivo en /root/Startup.
2. Pega el script.
3. Ejecuta chmod +x Zram-swap.sh.
4. Reinicia.
5. Comprueba con free -h y swapon --show.

```
#!/bin/sh
echo "Activando ZRAM swap..."
modprobe zram
echo lzo-rle > /sys/block/zram0/comp_algorithm
echo 4G > /sys/block/zram0/disksize
mkswap /dev/zram0
swapon /dev/zram0
echo "ZRAM swap activado."
free -h
```

Cómo comprobarlo: Resultado esperado: aparece /dev/zram0 como swap activa y el sistema responde mejor si la RAM se aprieta.

bootlocal.sh

Mueve actividad temporal como /tmp, logs y parte de la caché del navegador a RAM.

Cómo aplicarlo paso a paso
1. Crea o edita bootlocal.sh.
2. Pega el código.
3. Dale permiso de ejecución.
4. Reinicia.
5. Comprueba /tmp y /var/log con mount.

```
#!/bin/sh
sysctl -w vm.swappiness=10
swapoff -a
if ! mount | grep -q "/tmp type tmpfs"; then
    mount -t tmpfs -o size=1G tmpfs /tmp
fi
mkdir -p /tmp/log
mount --bind /tmp/log /var/log
mkdir -p /tmp/browser_cache
```

Cómo comprobarlo: Resultado esperado: /tmp es tmpfs, los logs viven en /tmp/log y la actividad temporal toca menos el disco.

pp4mnk-sysctl-loader

Carga automáticamente los valores de sysctl en cada arranque.

Cómo aplicarlo paso a paso
1. Pon el script en /root/Startup.
2. Hazlo ejecutable.
3. Asegúrate de que sysctl.config o /etc/sysctl.conf contiene los valores.
4. Reinicia.
5. Comprueba con sysctl.

```
#!/bin/sh
# Apply sysctl settings at every boot
sysctl -p >/dev/null 2>&1
```

Cómo comprobarlo: Resultado esperado: los valores elegidos siguen ahí después de reiniciar.

sysctl.config

Guarda los valores reales del kernel para red y memoria.

Cómo aplicarlo paso a paso
1. Crea el archivo con esos valores.
2. Cópialo a /etc/sysctl.conf o cárgalo explícitamente.
3. Guarda.
4. Reinicia o ejecuta sysctl -p.
5. Verifica valores concretos con sysctl.

```
# Max buffers
net.core.rmem_max=67108864
net.core.wmem_max=67108864

# TCP buffers
net.ipv4.tcp_rmem=4096 87380 67108864
net.ipv4.tcp_wmem=4096 65536 67108864

# Congestion control
net.ipv4.tcp_congestion_control=cubic

# Stability and efficiency
net.ipv4.tcp_fastopen=3
net.ipv4.tcp_mtu_probing=1
net.ipv4.tcp_slow_start_after_idle=0

# Network queue
net.core.netdev_max_backlog=5000

# RAM tuning
vm.swappiness=0
vm.vfs_cache_pressure=25
vm.dirty_ratio=10
vm.dirty_background_ratio=5
```

Cómo comprobarlo: Resultado esperado: mejor manejo de buffers TCP, menos tendencia a swap y páginas sucias más controladas.

Aviso importante
El cargador mostrado ejecuta `sysctl -p`, que por defecto lee `/etc/sysctl.conf`. Si guardas tus valores en un archivo llamado `sysctl.config`, debes copiarlo a `/etc/sysctl.conf` o cambiar el cargador para que lea explícitamente ese archivo.

pp4mnk-network-optimize

Activa funciones de offload de la tarjeta de red y aumenta la cola de transmisión.

Cómo aplicarlo paso a paso
1. Instala ethtool si falta.
2. Crea el script.
3. Hazlo ejecutable.
4. Reinicia.
5. Comprueba con ethtool -k eth0 e ifconfig eth0.

```
#!/bin/sh
ethtool -K eth0 tso on
ethtool -K eth0 gso on
ethtool -K eth0 sg on
ifconfig eth0 txqueuelen 5000
```

Cómo comprobarlo: Resultado esperado: TSO, GSO y SG activados; txqueuelen a 5000.

pp4mnk-MTU-tunning

Aplica MTU 1492 después de un pequeño retraso al arrancar.

Cómo aplicarlo paso a paso
1. Crea el script.
2. Hazlo ejecutable.
3. Reinicia.
4. Comprueba con ip link show eth0.

```
#!/bin/sh
sleep 2
ip link set eth0 mtu 1492
```

Cómo comprobarlo: Resultado esperado: eth0 informa mtu 1492.

pp4mnk-dns-loader-v3

Reescribe resolv.conf con los DNS preferidos y opciones agresivas de resolución.

Cómo aplicarlo paso a paso
1. Crea el script.
2. Hazlo ejecutable.
3. Reinicia.
4. Mira /etc/resolv.conf.
5. Prueba con nslookup.

```
#!/bin/sh
RESOLVCONF="/etc/resolv.conf"
LOGFILE="/tmp/dns-loader.log"

cat > "$RESOLVCONF" <<EOF
nameserver 1.0.0.1
nameserver 1.1.1.1
nameserver 8.8.8.8
options timeout:1 attempts:2 single-request-reopen
EOF
```

Cómo comprobarlo: Resultado esperado: resolv.conf con 1.0.0.1, 1.1.1.1 y 8.8.8.8 y opciones de timeout cortas.

Aviso para principiantes
Algunos gestores de red vuelven a escribir `/etc/resolv.conf`. Si tus DNS cambian solos tras reconectar, revisa ese punto antes de pensar que el script no funciona.

powertop-autotune

Espera a que termine el arranque y aplica recomendaciones de ahorro energético.

Cómo aplicarlo paso a paso
1. Instala powertop si hace falta.
2. Crea el script.
3. Hazlo ejecutable.
4. Reinicia.
5. Comprueba los tunables de powertop.

```
#!/bin/sh
sleep 20
powertop --auto-tune
```

Cómo comprobarlo: Resultado esperado: varios tunables pasan a Good tras el arranque.

5. Comandos de verificación y problemas típicos

Qué revisar	Comando	Qué debería verse
Límite CPU	cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq	2000000 si usas el modo equilibrado
MTU	ip link show eth0	mtu 1492
DNS	cat /etc/resolv.conf	1.0.0.1 / 1.1.1.1 / 8.8.8.8 y opciones del resolver
sysctl	sysctl net.ipv4.tcp_fastopen	valor = 3
swap zram	swapon --show	Aparece /dev/zram0 como swap activa
tmp en RAM	mount | grep "/tmp type tmpfs"	/tmp montado como tmpfs
offloads	ethtool -k eth0	TSO/GSO/SG activados si el script funciona

- Si un cambio no aparece tras reiniciar, lo primero es revisar si el script tiene permiso de ejecución.
- Si los DNS no permanecen, probablemente otro servicio está reescribiendo resolv.conf.
- Si la red empeora con offloading, desactiva ese script y vuelve a medir.
- Si la swap zram no aparece, comprueba que el módulo zram está disponible en el kernel.
- Si /tmp ya estaba montado de otra manera, adapta bootlocal.sh en lugar de forzarlo a ciegas.

Conclusión para novatos
No es obligatorio aplicar todo. Lo más importante es entender la lógica: menos calor cuando no necesitas potencia máxima, menos escrituras en disco cuando puedes usar RAM, y ajustes de red persistentes que se puedan verificar con comandos sencillos. Con esa base ya obtienes la mayor parte del beneficio.